

System-Level Modelling and Simulation of MEMS-Based Sensors

Virk, Kashif Munir; Madsen, Jan; Shafique, Mohammad; Menon, Aric Kumaran

Published in:
9th IEEE International Multi-Topic Conference (INMIC)

Link to article, DOI:
[10.1109/INMIC.2005.334503](https://doi.org/10.1109/INMIC.2005.334503)

Publication date:
2005

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Virk, K. M., Madsen, J., Shafique, M., & Menon, A. K. (2005). System-Level Modelling and Simulation of MEMS-Based Sensors. In 9th IEEE International Multi-Topic Conference (INMIC) IEEE. DOI: 10.1109/INMIC.2005.334503

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

System-Level Modeling and Simulation of MEMS-based Sensors

Mohammad Shafique

Aric Menon

Micro/Nano Tribology & Modeling Group

Micro-Electro-Mechanical Systems (MEMS) Section

Department of Micro & Nano Technology

Technical University of Denmark, Lyngby 2800, Denmark

Email: {msq, am}@mic.dtu.dk

Kashif Virk

Jan Madsen

System-on-Chip Group

Computer Science & Engineering Section

Department of Informatics & Mathematical Modeling

Technical University of Denmark, Lyngby 2800, Denmark

Email: {virk, jan}@imm.dtu.dk

Abstract—The growing complexity of MEMS devices and their increased use in embedded systems (e.g., wireless integrated sensor networks) demands a disciplined approach for MEMS design as well as the development of techniques for system-level modeling of these devices so that a seamless integration with the existing embedded system design methodologies is possible.

In this paper, we present a MEMS design methodology that uses VHDL-AMS based system-level model of a MEMS device as a starting point and combines the top-down and bottom-up design approaches for design, verification, and optimization. The capabilities of our proposed design methodology are illustrated through the design of a microaccelerometer.

I. INTRODUCTION

Integrated Microsystems and their subset — MicroElectro-Mechanical Systems (MEMS) — are inherently complex in nature. The level of their complexity can be realized from the fact that these systems involve coupled energy domains (e.g., electrical, mechanical, magnetic, fluidic, optical, etc.) and their signal conditioning units typically involve continuous-time (analog) and discrete-time (digital) electronic domains or a mixture of both (mixed-signal). The prototyping of these systems, using the available manufacturing techniques is usually very expensive. Therefore, the existing "build-and-test" approach for these systems has to be replaced by a systematic design methodology that introduces design hierarchy and information sharing across the domain dichotomies. As a part of design methodology, modeling and simulation of MEMS-based systems play an important role in reducing the number of design iterations and their time-to-market.

The design methods of MEMS have traditionally been viewed from either a bottom-up or a top-down perspective.

In the bottom-up approach, which, presently, is the most common design approach among the MEMS design communities, the idea of a MEMS device is conceived and the necessary physical-level modeling on the device design is conducted to establish its physical characteristics. However, the computational resource requirements associated with physical-level modeling render it an impractical approach for modeling the entire system. Therefore, the physical-level modeling techniques are only employed to analyze the physical characteristics of MEMS device structures and to generate the

data necessary to create a reduced-order model of the device¹. The reduced-order modeling of the MEMS device, along with the necessary signal conditioning and control electronics, is then conducted to determine its proper functioning at the device level. System-level modeling is then carried out to determine the potential impact the device will have on the whole system.

On the other hand, in the top-down approach, the critical system parameters are first determined from the system-level (reduced-order) analytical equations governing the system behavior regardless of the implementation options or the process technology to be used. After determining the critical system parameters, the implementation details and the specific process technologies are considered through the use of device-level reduced-order models. Modeling at the device level involves a MEM structure with or without the signal conditioning and control electronics. At the device-level, reduced-order modeling allows the designers to determine what boundary and load conditions will be placed on individual components. After device-level modeling, more detailed physical-level modeling (3D Modeling) allows the designer to examine a structure's response to a particular physical environment in finer detail.

In this paper, we mainly focus on system-level modeling of MEMS-based sensors using VHDL-AMS as it supports multi-domain, mixed-signal modeling capabilities needed for system-level modeling of MEMS-based systems. We propose a model-driven MEMS-based system design methodology in which the design specification is captured in a system-level model using VHDL-AMS which is subsequently refined in a step-by-step manner to yield the physical design. By back-annotating the refined design parameters obtained through physical-level simulations, the same specification-based system-level model can be used for system optimization through design-space exploration by iterating back and forth the design hierarchy until a fully optimized system design is achieved. Our proposed design methodology can be viewed as a combination of top-down and bottom-up design approaches (described above) which have been modified to allow for optimization through design-space exploration.

¹The techniques associated with reduced-order modeling are essentially the same as macro-modeling.

The rest of this paper is organized as follows: Section II provides an overview of the current research in the field of system-level modeling of integrated microsystems. Section III gives a detailed explanation of our proposed model-driven design methodology. An illustrative example elaborating the capabilities of our proposed design methodology applied to the system-level modeling and design of a microaccelerometer for wireless sensor network applications is presented in Section IV. Section V, finally, provides conclusions and the future directions of our work.

II. RELATED WORK

HDL's² have been used since the 1960s to model and simulate applications as diverse as (digital and analog) electronic systems and fluid concentrations in chemical processes. Modern HDLs support the description of both behavior and structure.

Depending on the available language constructs, HDLs can be divided into digital, analog, and mixed-signal HDLs. Digital HDLs, such as VHDL or Verilog, are based on event-driven modeling techniques and use a discrete model of time. They support the modeling of digital hardware at abstraction levels, ranging from system level down to the device level. Analog HDLs support the description of systems of differential and algebraic equations (DAEs) whose solution varies continuously with time. Analog HDL's like Verilog-A, MAST, VHDL-AMS, etc. support multi-domain and mixed-signal modeling capabilities and have been effectively used for MEMS modeling. HDL-A and MAST are the proprietary languages, so their development is vendor-dependent. On the other hand, Verilog-AMS and VHDL-AMS are open-source languages.

VHDL-AMS is an informal name for a combination of two IEEE standards: VHDL 1076-1993 and VHDL 1076.1-1999. It covers most of the modeling requirements for MEMS, and sufficient work (e.g., [1]–[3]) has been done to model electromechanical, MOEMS, fluidic, magnetic, and thermal systems. VHDL-AMS supports hierarchical description of continuous, mixed-domain, and discrete, conservative and non-conservative physical systems. An overview on modeling conservative systems with analog and mixed signal is discussed in [4], [5]. At the device level, VHDL-AMS has been effectively used to predict the behavior of interacting energy domains (e.g., magnetic, mechanical, electrical, etc.) using an integrated, multiple-domain, system representation.

Several techniques have been reported for the system-level modeling of MEMS devices. At the system level, MEMS devices are modelled as lumped-parameter elements (spanning multiple energy domains) along with associated electronics (analog, mixed-signal, digital). The equations governing the device behavior are the Ordinary Differential Equations (ODEs) and the Difference Equations (DEs). A hierarchically-structured design methodology for designing suspended MEMS devices which is compatible with the

standard mixed-signal ASIC design flow has been described in [6], [7] using the example of a microresonator and a microaccelerometer. The design approach takes advantage of parameterized component libraries for device layout generation and modern analog and mixed-signal hardware description languages such as VHDL-AMS and Verilog-A which allow the use of non-electrical energy domains for behavioral or system-level simulation of the MEMS device. The design methodology also provides design-space exploration capability.

In [8], the general aspects concerning the design automation for microsystems (in particular, MEMS devices) are considered but no clear design methodology has been presented. In [9], a modular design methodology for suspended MEMS has been presented that uses circuit-level behavioral simulation, schematic-driven layout generation, and system-level simulation in VHDL-AMS and Verilog-AMS. A comprehensive, multi-domain, multi-language system-level modeling of Systems-on-Chip embedding MEMS devices (using SystemC and VHDL-AMS) has been reported by [10] but it does not complement the system-level model with a concrete and workable MEMS design methodology.

The top-down design of MEMS and the underlying design challenges have been discussed by [11], [12] whereas [13] discuss the bottom-up design methodology and system-level modeling using VHDL-AMS. [14] uses top-down design approach for device design and bottom-up approach for design verification. In this paper, we extend this approach to include design optimization through back annotation of refined device parameters obtained by FEM analysis at the physical level into the system-level model described in VHDL-AMS.

III. MODEL-DRIVEN MEMS DESIGN METHODOLOGY

As mentioned above, to specify, design, and implement a complex MEMS-based sensing device, it is modeled at four levels of abstraction: process-level, physical-level, device-level, and system-level. Physical-level modeling involves numerically solving the equations of physics governing the system behavior using numerical solvers such as the Finite-Element Method (FEM), Boundary Element Method (BEM), etc. Device-level modeling involves reduced-order modeling through the generation of macro-models from the physical-level models using the macro-modeling solvers. System-level modeling techniques involve block diagram-based system representation (e.g., Simulink) or it may also involve reduced-order modeling using HDL's (SystemC-AMS, VHDL-AMS, etc.) or Parametric Design Libraries³, etc. We propose a model-driven MEMS design methodology (see Figure 1) that supports component-based design accompanied by substantial component reuse.

Starting from a design concept, a system-level model of a MEMS device is constructed for functional simulation of the design concept and for design-space exploration by examining and changing the behavioral and performance characteristics of the design concept till it meets the desired system specifications.

³For example, in the Coventor design environment, a 3D model of a MEMS device can also be generated from a parametric design library-based model.

²Hardware Description Languages (HDL's) are the programming languages specifically designed for describing the behavior of physical devices and processes. Models written in an HDL are used as input to a suitable simulator to analyze the behavior of the devices.

The key design parameters are extracted from the system-level model to construct a reduced-order model at the device level. The schematic-based, circuit-level reduced-order models involve lumped-parameter device models having few degrees of freedom with analytical or semi-analytical equations describing the behavior of the components comprising the MEMS device (e.g., beams, plates, combs, etc.). These models are written in an HDL and compiled into a design library called the parameterized design library. A MEMS device is composed by connecting together the required components. These device models can analyze a complex device behavior in a very short time.

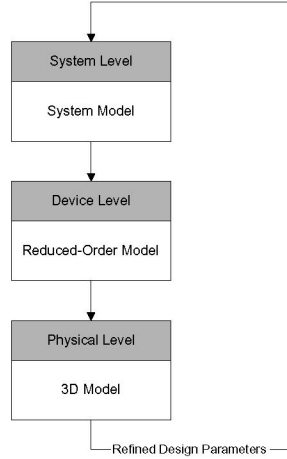


Fig. 1. System-Driven MEMS Design Methodology

After reduced-order modeling, a 3-D device structure is obtained at the physical level which is subjected to FEM-based analysis to extract the refined design parameters which are back-annotated into the system-level model for design refinement. This process is iterated till a final optimized device design is obtained.

IV. ILLUSTRATIVE EXAMPLE: MICROACCELEROMETER MODELING & DESIGN

To illustrate the capabilities of our proposed MEMS design methodology, a system-level model for a capacitive microaccelerometer has been developed at the system level using VHDL-AMS which allows direct simulation of mechanical and analog and digital electric sub-systems in their respective domains without any analogy transformation.

If the proof mass of the sensing element of a microaccelerometer has a mass of M , the suspension beams have an effective spring constant of K , and there is a damping factor D , affecting the dynamic movement of the mass, the microaccelerometer can be represented by a second-order mass-damper-spring system (see Figure 2). If an external force, F , displaces the support frame relative to the proof mass, the internal stress in the suspension spring changes. Both, this relative displacement and the suspension-beam stress can be used as a measure of the external force. By using Newton's second law [15]:

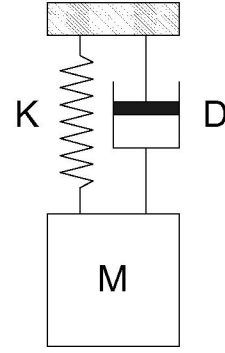


Fig. 2. Lumped-Parameter Model of Microaccelerometer

$$M\ddot{x} + D\dot{x} + Kx = F$$

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = a$$

In the above equation, x is the proof mass displacement, $\omega_n = \sqrt{\frac{K}{M}}$ is the natural resonance frequency, $\zeta = \frac{D}{2M\omega_n}$ is the damping factor, and $Q = \sqrt{\frac{KM}{D}}$ is the quality factor,

The resonance frequency, ω_n , of the structure can be increased by increasing the spring constant, K , and decreasing the proof mass, M , while the quality factor, Q , of the device can be increased by reducing the damping factor, D , and by increasing the proof mass, M , and spring constant, K .

The static sensitivity of the microaccelerometer is:

$$S_{static} = \frac{x_{static}}{a} = \frac{1}{\omega_n^2}$$

The static response of the device, i.e., its static sensitivity, S_{static} , can be improved by reducing its resonant frequency, ω_n .

The primary mechanical noise source for the device is due to the Brownian motion of the gas molecules surrounding the proof mass and the Brownian motion of the proof mass suspension or anchors. The total noise equivalent acceleration (TNEA) is:

$$TNEA = \frac{\sqrt{4k_bTD}}{M} = \sqrt{\frac{4k_bT\omega_n}{QM}}$$

where,

k_b is the Boltzmann's constant and T is the absolute temperature.

Thus, to reduce the mechanical noise, the quality factor, Q , and the proof mass, M , have to be increased.

For the system response to be linear, a , has to be less than $0.3\omega_n$. This sets the maximum bandwidth, B_{max} , of the microaccelerometer.

$$2\pi B_{max} \leq 0.3\omega_n \Rightarrow \omega_n \geq 20.944 B_{max}$$

For the least amplitude distortion and for the output to follow the input over the widest input frequency range, the system has to be critically damped. This implies that:

$$\zeta = 0.707$$

or

$$\frac{D}{2M\omega_n} = 0.707$$

The minimum detectable acceleration can be obtained from the expression for TNEA:

$$a_{min} = \frac{\sqrt{4k_b T D}}{M} = \sqrt{\frac{4k_b T \omega_n}{Q M}}$$

The maximum detectable acceleration can be obtained from $S_{dynamic}$ as:

$$a_{max} = \frac{K g_{max}}{M}$$

However, g_{max} cannot be below $\frac{g}{3}$, because after that pull-in voltage ($V_p = \frac{2}{3} \sqrt{\frac{2Kg}{A\epsilon}}$) kicks in and electrodes collide with each other.

The difference between a_{max} and a_{min} gives the dynamic range of the microaccelerometer. There is a big tradeoff between dynamic range and sensitivity.

In the most general case, the proof-mass motion can have six degrees of freedom. But, typically, in a unidirectional accelerometer, the geometrical design of the suspension is such that one of these is dominant and the device has low off-axis sensitivity. The cantilever support has been one of the early popular suspension support designs, due to its simplicity, lower spring constant, and internal stress relief of the beams. However, this configuration results in a larger off-axis sensitivity unless the device is fully symmetric. Symmetric, full-bridge supports result in a very low off-axis sensitivity. By using a crab-leg or folded-beam configuration in a full-bridge support, the residual stress of the beams can also be relieved keeping the spring constant unchanged due to tensile and compressive stresses. The spring constant for the folded beam configuration employing straight truss is:

$$K = \left(\frac{\pi^4}{6}\right) \left[\frac{Ewh^3}{(2L_1)^3 + (2L_2)^3} \right]$$

If parallel-plate estimates are used to get the correct order of magnitude of capacitances:

$$C_{sense} = N \frac{\epsilon_0 h l}{g_0 \pm x}$$

The plates of a parallel-plate capacitor attract each other with an electrostatic force of:

$$F = \frac{\epsilon_0 h l}{2g_0^2}$$

The mass of the proof mass with attached cantilever electrodes can be estimated from the device dimensions.

The damping factor is a difficult quantity to calculate because the effect of *squeezed-film* damping between the fingers must be added to the *Couette flow* beneath the proof mass as it displaces. Further, if the aspect ratio of the air gaps between the fingers is low, even squeezed-film damping estimates are inaccurate. The damping factor, D_1 , obtained from the Couette flow⁴ is:

$$D_1 = \eta \frac{A}{h}$$

where,

η is the viscosity of the surrounding air/gas. The damping factor, D_2 obtained from the squeezed-film model is:

$$D_2 = N \eta \frac{lw^3}{\pi^4 h_0^3}$$

The total damping, D , is the sum of D_1 and D_2 .

The differential capacitors have high sensitivities and can be configured to give a linear response and are, therefore, preferred for many applications. The differential capacitors have the virtue of cancelling many effects to first order, providing a signal that is zero at the balance point and carries a sign that indicates the direction of motion. From the system point of view, a differential capacitor accomplishes linearization about the balance point. Consider an interdigitated parallel-plate differential capacitor with the gap of the upper capacitor g_1 and that of the lower capacitor g_2 . Assuming an equal area of both capacitors, a voltage $+V_s$ is applied to the upper plate and a voltage, $-V_s$ is simultaneously applied to the lower plate. The voltage appearing at the voltage divider output is:

$$V_0 = -V_s + \frac{C_1}{C_1 + C_2} 2V_s = \frac{C_1 - C_2}{C_1 + C_2} V_s$$

since the areas are equal:

$$V_0 = \frac{g_1 - g_2}{g_1 + g_2} V_s$$

If the two gaps are equal, the output voltage is zero. However, if the middle plate moves so that one gap is larger than the other, the output voltage is a linear function of this change. The resulting sensing element output is a square wave with amplitude proportional to the displacement and, hence, the acceleration magnitude. The phase of the output square wave relative to the excitation determines the acceleration polarity which measures the unbalance in the differential capacitor. The output is amplified, synchronously demodulated, low-pass filtered and digitized with a $\Sigma - \Delta$ A/D converter to give the output value. Since the demodulator is phase synchronized with the excitation signal, the output signal polarity correctly indicates the direction of the applied acceleration [16].

```
library ieee;
use ieee.math_real.all;
use ieee.mechanical_systems.all;
use IEEE.electrical_systems.all;
use work.all;

entity testbench is
end entity testbench;

architecture ideal of testbench is
terminal telect1, telect2 : electrical;
terminal ttrans1, ttrans2 : translational;
quantity tq : real;
terminal tinput1, tinput2 : electrical;

begin
  ins_spring: entity work.spring(linear)
  port map (trans1 => ttrans1, trans2 => ttrans2);

  ins_mass: entity work.mass(ideal)
  port map (trans1 => ttrans1, trans2 => ttrans2, elect1 => tinput1,
    elect2 => tinput2, q_v => tq);

  ins_damper: entity work.damper(ideal)
  generic map (D => 0.0)
  port map (trans1 => ttrans1, trans2 => ttrans2);

  ins_parallel_plate: entity work.parallel_plate(ideal)
  port map (elect1 => telect1, elect2 => telect2, trans1 => ttrans1,
    trans2 => ttrans2, q_c => tq);

  ins_v_pulse: entity work.v_pulse(ideal)
  generic map (pulse => 5.0, width => 50ms, period => 100ms)
  port map (pos => telect1, neg => electrical_ref);

  ins_forcepulse: entity work.forcepulse(ideal)
  generic map (pulse => 1000.0, width => 500ms, period => 1000ms)
  port map(trans_pos => ttrans1, trans_neg => translational_ref);
end architecture ideal;
```

Fig. 3. Top-level VHDL-AMS Code for System-level Model

⁴Steady viscous flow between parallel plates, one of which is moving parallel to the other, is called Couette flow

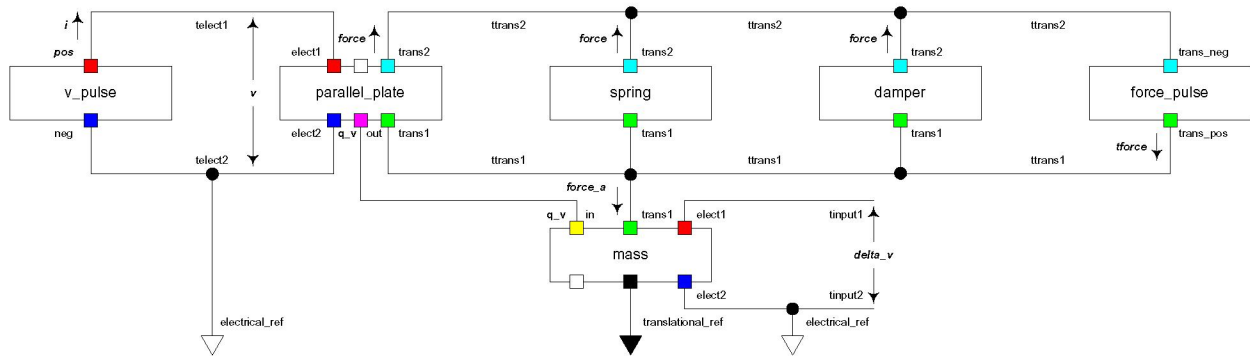


Fig. 4. System-Level VHDL-AMS Model Block Diagram

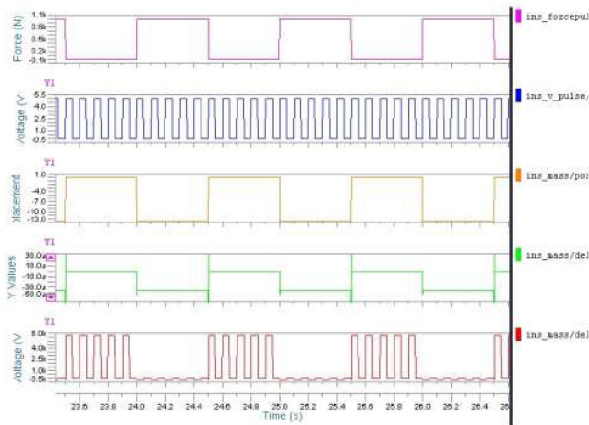


Fig. 5. System-Level Transient Simulation in VHDL-AMS

A surface micromachined, single-axis, lateral capacitive microaccelerometer sensing element consists of a movable beam (seismic mass), suspended by two spring tethers on either end. Movable fingers are attached to the mass. The fingers establish, together with fixed plates, capacitances that are evaluated by an electronic circuit. If the seismic mass is moved by an external force, the capacitances depend on this force. The structure is highly regular. Similar microelectromechanical devices are used in force-balanced accelerometers like the ADXL series from Analog Devices and Siemens.

Using the proposed model-driven design methodology, general system-level modeling of the microaccelerometer has been performed in VHDL-AMS using the above equations, as well as the mechanical relations describing the spring constant and the damping factor as a function of the device geometry and the ambient pressure. The basic functionality of the microaccelerometer has been simulated using VHDL-AMS (see Figure 3) A block diagram description of the VHDL-AMS model is given in Figure 4. The time-domain characteristics are simulated for different modeling approaches using simulation and are shown in Figure 5.

Further, the first-order device design optimization has been performed using the same equations, while the final microaccelerometer sensing element design has been simulated and optimized using the commercially-available finite-element

TABLE I
DESIGN PARAMETERS

Proof Mass	
Length	410 μ
Width	90 μ
Height	10 μ
Damping Holes	
Length	10 μ
Width	10 μ
Depth	10 μ
Number	80
Finger	
Length	160 μ
Width	10 μ
Thickness	5 μ
Number	22
Suspension Support	
Length1	170 μ
Length2	185 μ
Length3	30 μ

method (FEM) solvers in the Coventor software package.

Figure 6 shows the schematic drawing of the sensing element and the electronic circuitry of such a microaccelerometer built using components from the parametrized design library of Saber and MAST in the Coventor design environment. The device-level model consists of mechanical beams of different dimensions. The electrostatic forces are modeled by comb models (see Figure 7). The design parameters are listed in Table I. The advantage of this (schematic-based) approach is an easy combination of these MEMS primitives with other user-defined models. Therefore, for functional simulation, behavioral models of the electronic subsystems were used. Figure 8 shows the displacement of the sensing element at the first resonance frequency.

V. CONCLUSIONS

We have proposed a model-driven MEMS design methodology which is more than a combination of the existing top-

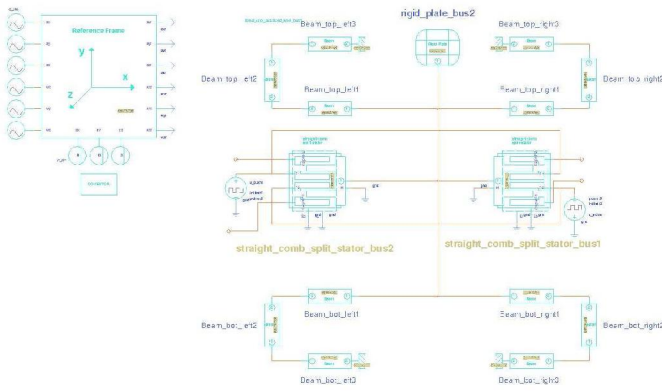


Fig. 6. Sensing Element Schematic

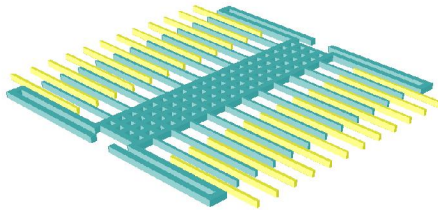


Fig. 7. Sensing Element

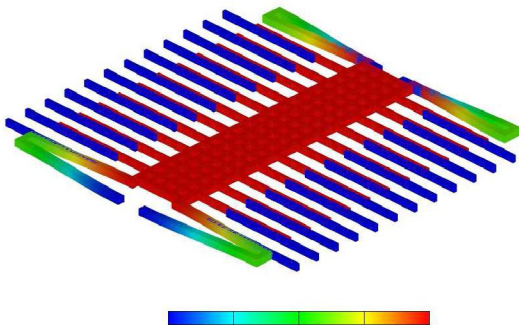


Fig. 8. Sensing Element FEM Simulation in Coventor

down and bottom-up design approaches as it enables MEMS design, validation, and optimization in a consistent, step-by-step manner and is compatible with the existing embedded system design methodologies. We have illustrated the capabilities of our proposed MEMS design methodology by applying it to design, simulate, and optimize a microaccelerometer sensor. System-level modeling of MEMS-based sensors is an

ongoing research area with the aim to model, as accurately as possible, the microsystem device behavior at the system level as well as at the lower levels of abstraction. The capabilities of the existing design tools for designing microsystems are still limited to some extent, either in the diversity of components in the parametric components library, accuracy of simulation, or in terms of simulation speed and ease of use. Lack of standards and interoperability are additional limitations. Information about macromodeling or lumped parameter modeling of microsystems is ample but patchy. Moreover, automated synthesis of microsystems from system-level models seems to be a far-off dream. The work described here gives a good insight into the system-level modeling of microsystems which can stimulate ideas about hybrid systems modeling and verification especially in the context of the emerging new area of wireless integrated sensor networks.

REFERENCES

- [1] F. Pecheux, C. Lallement, and A. Vachoux, "Vhdl-ams and verilog-ams as alternative hardware description languages for efficient modeling of multidiscipline systems," *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on*, vol. 24, no. 2, pp. 204–225, 2005.
- [2] P. Wilson, J. Ross, A. Brown, and A. Rushton, "Multiple domain behavioral modeling using vhdl-ams," *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, pp. V-644, 2004.
- [3] S. Guessab and J. Oudinot, "Modeling of a piezoelectric device with shocks management using vhdl-ams," *Behavioral Modeling and Simulation Conference, 2004. BMAS 2004. Proceedings of the 2004 IEEE International*, pp. 19–24, 2004.
- [4] Christen, Ernst and Bakalar, Kenneth, "VHDL 1076.1 - analog and mixed-signal extensions to VHDL," *European Design Automation Conference - Proceedings*, pp. 556–561, 1996.
- [5] Ernst Christen and Kenneth Bakalar, "VHDL-AMS - A Hardware Description Language for Analog and Mixed-Signal Applications," *IEEE Transactions on Circuits and Systems*, vol. 46, no. 10, pp. 1263–1272, October 1999.
- [6] T. Mukherjee and G. K. Fedder, "Structured design of microelectromechanical systems," *Proceedings of Design Automation Conference (DAC)*, pp. 680–685, 1997.
- [7] T. Mukherjee, G. K. Fedder, and R. Blanton, "Hierarchical design and test of integrated microsystems," *IEEE Design and Test of Computers*, vol. 16, no. 4, pp. 18–27, 1999.
- [8] J. van Kuijk, G. Schropfer, and M. Dasilva, "Design automation for mems/mst," *Coventor BV. The Netherlands*, 2004.
- [9] T. Mukherjee, "Mems design and verification," *IEEE International Test Conference (TC)*, pp. 681–690, 2003.
- [10] Z. Juneidi, K. Torki, S. Martinez, G. Nicolescu, B. Courtois, and A. Jeraya, "Global modeling and simulation of system-on-chip embedding mems devices," *ASIC, 2001. Proceedings. 4th International Conference on*, pp. 666–669, 2001.
- [11] G. K. Fedder, "Top-down design of mems," *2000 International Conference on Modeling and Simulation of Microsystems - MSM 2000 and 2000 International Conference on Modeling and Simulation of Microsystems - MSM 2000*, pp. 7–10, 2000.
- [12] M. S. McCorquodale, F. H. Gebara, K. L. Kraver, E. D. Marsman, R. M. Senger, and R. B. Brown, "A top-down microsystems design methodology and associated challenges," pp. 20 292–20 296, 2003.
- [13] P. Schwarz and P. Schneider, "Model library and tool support for mems simulation," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4407, pp. 10–23, 2001.
- [14] Joachim Haase and Jens Bastian and Sven Reitz, "VHDL-AMS in MEMS Design Flow," in *Proceedings of Forum on Specification and Design Languages*, September 2002.
- [15] Senol Mutlu, University of Michigan, Ann Arbor, "Surface Micromachined Capacitive Accelerometer with Closed-Loop Feedback," <http://www.eecs.umich.edu/~smutlu/projects.html>, pp. 1–11.
- [16] Howard Samuels, "Single- and Dual-Axis Micromachined Accelerometers," *Analog Dialogue*, vol. 30, no. 4, pp. 3–5, October-December 1996.